# SEL0614 - Microprocessadores e Aplicações - Lista 1

Marcos Vinícius Firmino Pietrucci
10770072

**1- Fazer um programa que utilize um timer interno do 8051 para criar um tempo de atraso de 0.05 segundos. Utilizando este programa como uma sub-rotina, escrever um programa que atrase 5 segundos.**

```
                ORG 0000H
                SJMP SETUP

                ORG 000BH ;Interrupção do timer 0
                SJMP TC0_subrotina

SETUP:          SETB    EA
                SETB    ET0
                MOV     TMOD, #1H ;Software, temporizador modo 1
                MOV     TH0, #03CH     ; TC0 = 0.05s
                MOV     TL0, #0B0H

                MOV     R0, #100D    ; Serão 100 interrupcoes do TC0
                SETB    20H.0            ; Bit de controle

                SETB TR0


;***********Programa de 5s*****************************************************

LOOP:           JB    20H.0, $     ; Aguarda imterrupcao do TC0
                DEC    R0
                SETB    20H.0
                CJNE    R0,#0H, LOOP

                SJMP    $        ; Fim lógico do programa

;*******************************************************************************
TC0_subrotina:
                CLR 20H.0
                MOV     TH0, #03CH     ; TC0 = 0.05s
                MOV     TL0, #0B0H
                RETI
END
```

**2- Um sistema baseado no 8051 utiliza as duas interrupções externas disponíveis e ainda a interrupção gerada por 1 dos Temporizadores/Contadores. As condições em que se pretende que o sistema funcione são as seguintes:**

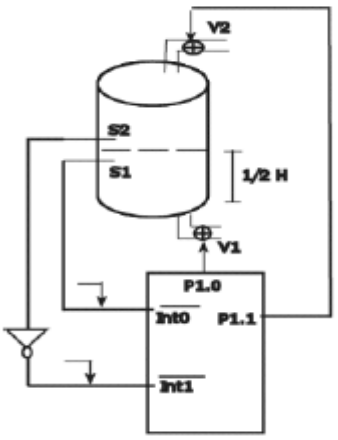
- A interrupção externa 0 deve ser semrpe atendida imediatamente e deve copiar o que está na posição de RAM externa 4000H para a posição 4200H
- A interrupção externa 1 deve escrever o que está em 4200H na porta P1
- A interrupção gerada pelo timer deve executar uma rotina que copie o que está na porta P2 para a posição 4000H da RAM externa
- No caso de duas interrupções acontecerem simultaneamente, deve ser atendida a interrupção externa.

```
                ORG 0000H
                SJMP SETUP

                ORG 0003H
                SJMP EXT_0

                ORG 000BH
                SJMP TC0

                ORG 0013H
                SJMP EXT_1

SETUP:          SETB    EA          ;Habilitar as interrupcoes
                SETB    ET0
                SETB    EX0
                SETB    EX1
                MOV     TMOD, #1H    ; Software, temporizador modo 1
                SETB    IT0
                SETB    IT1
                SETB    PX0         ; IExt 0 e 1 de alta prioridade
                SETB    PX1
                CLR     PT0         ; TC inter. de baixa prioridade
                MOV     TH0, #0FFH    ; TC0 = 0.05s
                MOV     TL0, #006H
                SETB    TR0

                SJMP    $
;*********************************************************
EXT_0:          CLR     EA      ; Previne que seja interrompida
                MOV     DPTR, #4000H
                MOVX    A,@DPTR
                MOV     DPTR, #4200H
                MOVX    @DPTR,A
                SETB    EA
                RETI
```

```
;*******************************************************
EXT_1:
                    MOV     DPTR, #4200H
                    MOVX    A,@DPTR
                    MOV     P1, A
                    RETI

;*******************************************************
TC0:
                    MOV     A, P2
                    MOV     DPTR, #4000H
                    MOVX    @DPTR, A
                    RETI

END
```

---

**3- Considere o Controlador de Nível da figura operando da seguinte maneira:**



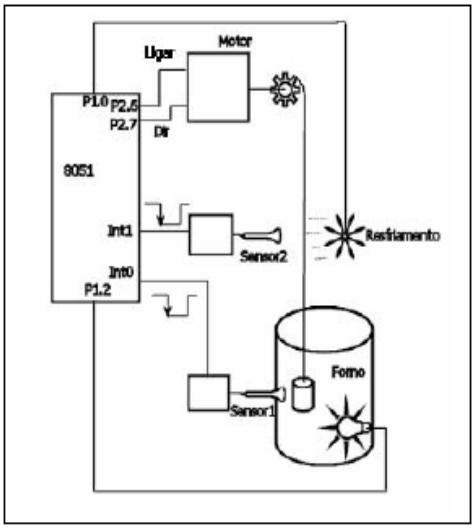

- Dois sensores S1 e S2 emitem nível lógico zero se estiverem fora do líquido e nível lógico 1 se estiverem imersos no líquido.
- Uma válvula V1, acionada pelo bit P1.0 de um microcontrolador 8051 drena o reservatório e uma válvula V2 acionada pelo bit P1.1 enche-o com líquido.
- Inicialmente o reservatório está vazio, ou seja, com os dois sensores em nível lógico zero.

```
                    ORG 0000H
                    SJMP SETUP

                    ORG 0003H
                    SJMP EXT_0_S1

                    ORG 0013H
                    SJMP EXT_1_S2

SETUP:              ;Configurar as interrupções
                    SETB EA
                    SETB EX0
                    SETB EX1
                    SETB IT0
                    SETB IT1
                    SETB PX0
                    SETB PX1
```

```
                ;O tanque começa vazio enchendo
                SETB    P1.1    ;V2
                CLR     P1.0    ;V1

                SJMP    $

;*********************************************************
EXT_0_S1:   ; Passou por S1, encher o tanque
                SETB    P1.1    ;V2
                CLR     P1.0    ;V1
                RETI

;*********************************************************
EXT_1_S2:   ; Passou por S1, esvaziar o tanque
                CLR     P1.1    ;V2
                SETB    P1.0    ;V1
                RETI
END
```

---

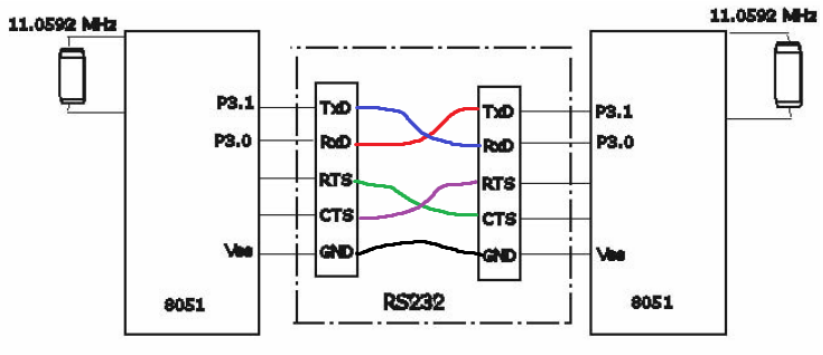
**4- Escrever um programa em Assembly do 8051 que controle o dispositivo de teste térmico de materiais, mostrado na figura. Um recipiente, com determinada substância sob teste, deve ser baixado (Dir = P2.7 = 1) através de um Motor (Ligar = P2.6 = 1), dentro de um forno. O Sensor1 detecta a presença do recipiente e envia uma descida de borda ao pino Int0 do microprocessador. O micro para o Motor (Ligar = P2.6 = 0) e aciona o aquecimento do forno (P1.2 = 1) durante aproximadamente 500 ms. Desliga o aquecimento, inverte o sentido do Motor (Ligar = P2.6 = 1) (Dir = P2.7 = 0) , erguendo o recipiente até a posição do Sensor2, que opera da mesma forma que o Sensor1, mas usando a Interrupção Int1. Quando Int1 receber uma descida de borda, o micro deve parar o Motor (Ligar = P2.6 = 0) e acionar o resfriamento (P1.0 =1) durante aproximadamente 500 ms. O ciclo deve ser repetido 3 vezes e parar. Considerar o Cristal da CPU de 12 MHz.**




```
                ORG 0000H
                SJMP SETUP

                ORG 0003H
                SJMP EXT_0

                ORG 0013H
                SJMP EXT_1

SETUP:          ;Configurar as interrupções
                SETB EA
                SETB EX0
                SETB EX1
                SETB IT0
                SETB IT1

                ;Inicia o teste, baixar o recipiente
                SETB    P2.7
                SETB    P2.6
                CLR     P1.2    ;Desliga aquecimento
                CLR     P1.0    ;Desliga resfriamento

                ;Contador de vezes
                MOV     R3, #0H

                ;Garantir que o processo ocorra 3 vezes
                CJNE    R3, #3H, $

                SJMP    $
```

```
;*******************************************************
EXT_0:      ;Presenca do recipiente para aquecer
            CLR   P2.6     ;Parar o motor
            SETB  P1.2           ;Liga aquecimento
            ACALL  Delay500
            CLR   P1.2     ;Desliga aquecimento
            CLR   P2.7     ;Inverte sentido do motor
            SETB  P2.6           ;Liga motor
            RETI

;*******************************************************
EXT_1:      ;Presença de recipiente para resfriar
            CLR   P2.6     ;Parar o motor
            SETB  P1.0           ;Liga resfriamento
            ACALL  Delay500
            CLR   P1.0     ;Desliga resfriamento
            SETB  P2.7           ;Inverte sentido do motor
            SETB  P2.6           ;Liga motor

            INC   R3       ;Incrementa o contador de vezes
            RETI

;*******************************************************

Delay500:
            ; START: Wait loop, time: 500 ms
            ; Clock: 12000.0 kHz (12 / MC)
            ; Used registers: R0, R1, R2
            MOV   R2, #004h
            MOV   R1, #0FAh
            MOV   R0, #0F8h
            NOP
            DJNZ  R0, $
            DJNZ  R1, $-5
            DJNZ  R2, $-9
            RET
            ; Rest: -13.0 us
END
```

**5- Dois microcontroladores 8051 estão se comunicando através de uma interface padrão RS232 com handshaking via RTS e CTS. No esquema abaixo conectar os fios do cabo de comunicação corretamente e responder aos itens:**




- Qual o valor de TH1 em ambos os micros se a taxa de comunicação é de 19200 *bauds*?

  Utilizando a fórmula, com um k = 2

$$TH1 = 256 - \frac{2 \times (110592 \times 10^{6})}{384 \times 19200}$$

$$TH1 = 253$$

- Qual o valor de tensão na linha de comunicação quando está em repouso?

  O padrão RS232 utiliza lógica negativa, por isso o nível lógico baixo são para tensões entre -25V e -3V.

  Além disso, comunicação em repouso é quando o nível lógico permanece em baixo, por isso o valor de tensão na linha de comunicação quando está em repouso é: entre -25V e -3V.

- Qual o tamanho em microsegundos do *Start Bit*?

  Dividindo o tempo 1 bit pela taxa de *bauds*

$$Sb = \frac{1}{19200} = 52\ \mu s$$

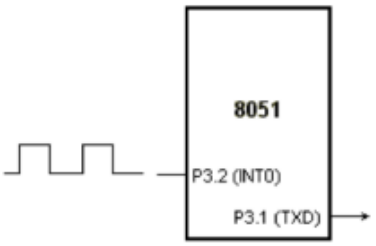
**6 - Um sistema baseado no Microcontrolador 8051 utiliza as duas interrupções externas e as interrupções geradas pelos Timers/Counters. Escrever um programa em Assembly tal que:**

- A interrupção externa 0, atendida prioritariamente, deve trocar o que está na Porta P1 com o que está contido na posição de RAM externa 5000h.
- A interrupção externa 1, com baixa prioridade, deve transferir o que está armazenado na RAM externa do endereço 5000h para a RAM interna no endereço 7Fh;
- A interrupção gerada pelo T/C 0 (a cada 10 ms) com alta prioridade, deve executar uma rotina que copie o que está no endereço da RAM interna 7Fh para a posição 5200h da RAM externa;
- A cada 60 milissegundos (aproximadamente) e controlado pelo T/C 1 com interrupção
- de baixa prioridade, o dado armazenado na RAM externa no endereço 5200h deve ser enviado para a posição de memória externa 5000h.

```
        ORG     0000H
        SJMP    SETUP

        ORG     0003H
        SJMP    EXT_0

        ORG     000BH
        SJMP    TC0

        ORG     0013H
        SJMP    EXT_1

        ORG     001BH
        SJMP    TC1

SETUP:  ;Configurar as interrupções
        SETB    EA
        SETB    EX0
        SETB    EX1
        SETB    ET0
        SETB    ET1
        SETB    IT0
        SETB    IT1
        MOV     TMOD, #22H    ;Modo dos contadores

        ;Prioridades
        SETB    PX0
        CLR     PX1
        SETB    PT0
        CLR     PT1
```

```
                ;Valores dos contadores
                MOV    TH0, #0D8H
                MOV    TL0, #0F0H
                MOV    TH1, #015H
                MOV    TL1, #0A0H

                SETB     TR0
                SETB     TR1

                SJMP    $

;*********************************************************
EXT_0:          MOV    DPTR, #5000H
                MOV    R0, P1
                MOVX    A, @DPTR
                MOV    P1, A
                MOV    A, R0
                MOVX    @DPTR, A
                RETI

;*********************************************************
EXT_1:          MOV    DPTR, #5000H
                MOVX    A, @DPTR
                MOV    7FH, A
                RETI

;*********************************************************
TC0:
                MOV    A, 7FH
                MOV    DPTR, #5200H
                MOVX    @DPTR, A
                RETI

;*********************************************************
TC1:
                MOV    DPTR, #5200H
                MOVX    A,@DPTR
                MOV    DPTR, #5000H
                MOVX    @DPTR, A
               RETI

END
```

**7- Fazer um programa em Assembly do 8051 que calcule o valor da frequência de uma onda quadrada entrando pelo pino da Interrupção Externa 0 e envie-o o para a interface serial RS232 a uma taxa de 4800,N,8,1. Considerar o cristal da CPU de 11,0592 MHz. Utilizar a interrupção Int0 sensível à descida de borda. O valor da frequência a ser enviada para a interface serial é um número hexadecimal de 16 Bits.**




```
                ORG     0000H
                SJMP    SETUP

                ORG     0003H
                SJMP    EXT_0

                ORG     000BH
                SJMP    TC0


SETUP:          ;Configurar as interrupções
                SETB    EA
                SETB    EX0         ;Conta os pulsos
                SETB    ET0         ;Conta tempo para frequencia
                SETB    IT0

                ;Zera acumulador, que contará os pulsos
                MOV     A, 0

                ;Timer 0 conta 10ms => Serão 100 vezes
                MOV     TH0, #0D8H
                MOV     TL0, #0F0H
                MOV     R0,  #0H    ;Contador de vezes -> Chegar em 1s

                ;Freq = 1/s = numero_pulsos em 1s

                ;Configura o contador 0
                MOV     TMOD, #1H   ;TC0 Modo 1

                SETB    TR0
                SJMP    $

;*****************************************************
ENVIA_SERIAL:   ;Desabilita as interrupcoes indesejadas
                CLR     EX0
```

```
                    CLR    ET0

                    ;Configura transmissao seriala
                    ;Timer 1 usado na transmissão serial
                    SETB   ET1
                    MOV    TMOD, #00100000b    ;TC1 modo 2
                    ;Calcular a taxa de comunicação
                    ;Pelos valores dados:
                    MOV    TL1, #250D    ;TH1 = 256 −(110592∗10^6)/(384∗4800) = 250
                    MOV   TH1, #250D
                    SETB   TR1

                    MOV   SCON, #40H ; Modo 1 do canal serial

                    MOV    SBUF, A       ;Transmite a frequencia

                    JNB   TI,$     ;Espera transmissao
                    CLR   TI

                    SJMP   $      ;Fim logico

;*********************************************************
EXT_0:              INC    A
                    RETI
;*********************************************************
TC0:                INC   R0
                    CJNE   R0,#64H,VOLTA    ;Contar 100 vezes interrupt de 10ms
                    SJMP   ENVIA_SERIAL
VOLTA:              MOV   TH0, #0D8H
                    MOV   TL0, #0F0H
                    RETI
;*********************************************************

END
```

---

**9 - Um robô como mostrado na figura é acionado por dois motores de corrente contínua, um para cada roda, conforme o esquema, e possui um sensor localizado na parte da frente que tem a função de detectar a presença de obstáculos. Desenvolver um programa em assembly do 8051 que controle o robô fazendo-o navegar por uma sala onde diversos obstáculos podem ser encontrados, de tal forma que ele não colida com nenhum.**

Especificações dadas no enunciado original

```
                ORG 0000H
                SJMP SETUP

                ORG 0003H
                SJMP EXT_0

SETUP:          ;Configurar as interrupções
                SETB    EA
                SETB    EX0
                SETB    IT0

                ;O robô começa andando para frente
                SETB    P1.0
                SETB    P1.1
                SETB    P1.2
                SETB    P1.3

                ;Bit de direção a virar
                CLR     20H.0

                SJMP    $

;*********************************************************
EXT_0:          CLR     EX0     ;Desabilita interrupt

                ;Marcha ré
                CLR     P1.1
                CLR     P1.3
                CALL    DELAY_2S
                JB      20H.0,Esq

Dir:            ;Virar a direita por 2s
                SETB    P1.1
                CLR     P1.3
                CALL    DELAY_2S
                SJMP    retorno

Esq:            ;Virar a esquerda por 2s
                CLR     P1.1
                SETB    P1.3
                CALL    DELAY_2S

retorno:        ;Voltar o robô para frente
                SETB    P1.1
                SETB    P1.3
                CPL     20H.0   ;Inverte direção de virar
```

```
                SETB    EX0
                RETI
;*******************************************************
DELAY_2S:
; START: Wait loop, time: 2 s
; Clock: 12000.0 kHz (12 / MC)
; Used registers: R0, R1, R2, R3
                MOV     R3, #003h
                MOV     R2, #0D2h
                MOV     R1, #00Ch
                MOV     R0, #082h
                NOP
                DJNZ    R0, $
                DJNZ    R1, $-5
                DJNZ    R2, $-9
                DJNZ    R3, $-13
                MOV     R1, #006h
                MOV     R0, #0BAh
                NOP
                DJNZ    R0, $
                DJNZ    R1, $-5
                NOP
                NOP
                NOP
                RET
; Rest: 0
; END: Wait loop

END
```

**10 - Automatizar uma Máquina de Doces com o Microcontrolador 89S52. A máquina deve fornecer em cada operação, somente um doce que custa 20 centavos. A cada operação o programa reinicia e espera nova sequência de moedas. As moedas aceitas pela máquina são de 5 centavos, 10 centavos e 20 centavos. Como cada moeda tem um tamanho diferente, um sensor óptico alinhado com o coletor de moedas determina qual moeda foi inserida. Apenas uma moeda pode ser inserida por vez. A inserção de uma moeda é detectada através da Interrupção Int0. O circuito de reconhecimento de moedas é mostrado abaixo e sua operação é de acordo com a Tabela 1.**

```
                ORG  0000H
                SJMP SETUP

                ORG  0003H
                SJMP EXT_0

SETUP:          ;Configurar as interrupções
                SETB    EA
                SETB    EX0
                SETB    IT0

                ;Maquina comeca zerada
RESET:          SETB    P1.0
                SETB    P1.1
                SETB    P1.2
                CLR     P2.0
                CLR     P2.1
                CLR     P2.2

                ;Armazena valor total inserido
                MOV     R0,#0H
                MOV     A, R0
LOOP:
                CJNE    A, #20D, TEST    ;Compara valor total inserido com 20
TEST:           JNC     Resposta

                SJMP    LOOP

Resposta:       ;Calcular se deve haver troco
                CJNE    A, #20D, TEST2
TEST2:          JC      LOOP
                MOV     A, R0
                CLR     C
                SUBB    A, #20D

                JNZ     tem_troco
```

```
                ;Se veio aqui não tem troco
                SJMP    FIM_result

               ;Verificar se a diferenca é 5
tem_troco:     CJNE    A, #5D, troco_nao_eh_5

                ;Se veio aqui o troco é 5
                SETB    P2.1
                SJMP    FIM_result

troco_nao_eh_5:    ;Verificar se a diferenca é 10
                CJNE    A, #10D, troco_nao_eh_10

                ;Se veio aqui o troco é 10
                SETB    P2.2
                SJMP    FIM_result

troco_nao_eh_10:
                ;Se o troco não é 5 nem 10 só pode ser 15
                SETB    P2.1
                SETB    P2.2
                SJMP    FIM_result

FIM_result:    SETB    P2.0    ;Dá o doce
               MOV    R0,#0H
               SJMP    RESET

;*********************************************************
EXT_0:          CLR    EA    ;Desabilita interrupt
                ;Detectar qual moeda foi inserida
                JB    P1.2, NaoEh20

                ;Se veio aqui eh 20
                MOV    R1, #20D
                SJMP    FIM_moeda

NaoEh20:       JB    P1.1, NaoEh10

                ;Se veio aqui eh 10
                MOV    R1, #10D
                SJMP    FIM_moeda

NaoEh10:       JB    P1.0, FIM_moeda

                ;Se veio aqui eh 5
                MOV    R1, #5D
                SJMP    FIM_moeda
```

```
                ;Se veio aqui eh nada
FIM_moeda:      SETB    EA
                MOV     A,  R0
                ADD     A,  R1
                MOV     R0, A
                RETI

END
```